REMARKS

This amendment is submitted in response to the Office Action dated June 21, 2007. Reconsideration and allowance of claims are requested. In this Office Action, claims 1-25 are considered. Claims 16, 19 and 23 are objected to as dependent on rejected claims, but are otherwise considered to recite allowable subject matter. Claims 1-15, 17, 18, 20, 22, 24 and 25 stand rejected. Specifically, claim 1 is rejected under 35 U.S.C. §102(e) as anticipated by *Puzak* (US 6,560,693) or *Kishi* (US 6,502,165). Claims 2-5 are rejected under 35 U.S.C. §103 as being unpatentable over *Puzak* considered with *Doherty* (US 6,115,083). Claims 1, 2, 6-8 and 24-25 are rejected under 35 U.S.C §102 as anticipated by *Gupta* (US 5,787,272). Claims 9 and 10 are rejected under 35 U.S.C. §103 as unpatentable under *Gupta* and further in view of *Yamasaki* (US 6,182,211). Claims 11 and 12 are rejected under 35 U.S.C. §103 as unpatentable under *Gupta* and further in view of *Nguyen* (US 7,013,382). Claims 13-15, 17 and 18 are rejected under 35 U.S.C. §102(e) as anticipated by *Lindholm* (US 7,015,913). Claim 20 is rejected under 35 U.S.C. §102 as anticipated by *Rishi* (US 5,953,530). Claims 21-23 are rejected under 35 U.S.C. §103(e) as being unpatentable over *Rishi* in view of *Cosgrove* (US 4,399,507). These rejections are respectfully traversed.

Considering first the rejection of claims 13-15, 17 and 18 as anticipated by *Lindholm*, independent claim 16 recites allowable subject matter over the reference. Therefore, claim 13 is amended to include claim 16. Claim 13 and its dependent claims 14, 15 and 18 are allowable.

The combination of claims 20 and 23 was indicated to comprise allowable subject matter. Therefore, these claims have been combined as claim 20. Claim 20 and its dependent claims 21 and 22 are therefore allowable, and such action is requested.

Claim 1 and its dependent claims have been amended to clarify that a multi threaded processing unit is provided which is capable of operating in either a divergent or non divergent mode. When operating in non divergent mode, the same instruction is applied to all samples of a group to optimize the speed of processing of multiple

samples. In divergent mode, one or more synch tokens is used so that a subset of the samples in the group can diverge from the other samples and be subjected to the operation of different instructions (such as call or return). At the end of the divergence, a synch token is detected, which is the instruction to perform synchronization on the divergent samples before proceeding to the next instruction in the program.

It is also possible that different members of the subset may re-synch with the entire group at different times by encountering different synch tokens. Therefore, after defining the diversion capability, the system detects a first synch token associated with one sample, looks for other synch tokens associated with other samples, and then proceeds to re-synch the members of the group which have encountered the first synch token. The re-synched members of the group can proceed to the next instruction.

Claim 2 has been amended to indicate that once the first synched token is encountered, the other samples of the subset must encounter the same synch token within a defined time period, or the effort to re-synch the group is abandoned.

These features do not appear in any of the references or any combination of the references. *Puzak* is cited against claim 1 and in combination with *Doherty* against claims 2-5. However, *Puzak* teaches nothing more than branching combined with storing the events that occur after such a branch so that this assumed sequence of events can then be followed. This is nothing more than a variation on a predictive execution of events and does not teach any of the recited elements noted above. *Doherty* does not overcome these deficiencies. The cited portions of *Doherty* only teach that two processors, which normally operate independently, are periodically stalled until both have the same instruction pending and can then be started on the same clock. *Doherty* does not teach anything about managing diverging graphic samples in a GPU, where a subset of the samples branches off and are then tested to see if these samples have encountered the same synch token within a defined period of time. As this combination of references is the only rejection of claims 3 and 5, these claims should be allowed along with their parent claims 1 and 2.

Claim 1 is also rejected as anticipated by *Kishi*. However *Kishi* only teaches a method of accessing a plurality of libraries of data that are stored separately. The system is taught in the context of a removable data storage system, with each storage

system being capable of supplying some updated synchronization token to the main processor which tracks the update level of the data volume see column 8, lines 5-23. Therefore, *Kishi* teaches nothing relevant to the independent claims as amended, and allowance of such claims, over *Kishi* is respectfully requested.

Gupta teaches a method and apparatus for synchronizing parallel processors, wherein the sequence of instructions is broken up into shaded and unshaded regions. The processors can only synchronize when they are both working in the shaded region. The unshaded regions are areas where the processors can never synchronize. The system depends entirely on a separate state machine 305 to read a want bit in each of the processors, and to coordinate synchronization among the processors who have signaled a "want." Gupta further teaches that each of the processors 201-204 must include a separate state machine where synchronization can only be accomplished when both processors are executing a sequence of code relevant to a shaded region. *Gupta* clearly fails to teach processing a subset of a group of samples, detecting that each sample of the group of samples has encountered a synch token and comparing the synch tokens to determine which of the divergent samples are to by synched prior to execution of the next instruction. Thererfore all of the claims 1, 2, 6-8, 24 &25 should be allowed.

The Examiner further relies on *Yamasaki* to reject claims 9 and 10. However, claim 9 clearly requires that each of the program counters corresponds to a different one of the group of samples, so that diverging samples of the group may execute different instructions. This claimed technique differs from *Yamasaki* who teaches a second program counter which saves an address of a subsequent instruction to a branch instruction. Thus, the program counters of *Yamasaki* are used to set up a sequence of instructions for execution, without a teaching of an assignment of each of a plurality of program counters to a separate sample among a group of divergent samples.

The Examiner also relies on *Nguyen* as teaching execution of routines of different depths. However, the claim further requires that each sample have associated with it a sub-routine depth indicating the number of sub-routines to be executed on that sample before synchronization is to occur the subroutine depth also determines the

order of execution of instructions on the associated sample. *Nguyen* does not teach association between a plurality subroutines and each sample of a group of samples, with the depth of the set of subroutines indicating the order of execution of instructions on the samples.

In view of these clear distinctions between the references cited and the claims as amended, reconsideration of and allowance of all the claims are respectfully requested submitted.

Respectfully submitted,

John C. Carey
Registration No. 51,530
PATTERSON & SHERIDAN, L.L.P.
3040 Post Oak Blvd. Suite 1500
Houston, TX 77056
Telephone: (713) 623-4844
Facsimile: (713) 623-4846
Attorney for Applicants